(19)日本国特許庁（ＪＰ）　(12) **公開特許公報（Ａ）**　(11)特許出願公開番号

特開平7－182368

(43)公開日　平成７年(1995)７月21日

(51)Int.Cl.[6]		識別記号	庁内整理番号	ＦＩ		技術表示箇所
Ｇ０６Ｆ	17/30					
	12/00	５４５　Ａ	8944－5B			
// Ｇ０６Ｆ	9/44	５５０　Ｎ	9193－5B			
			9194－5L	Ｇ０６Ｆ　15/ 401	３１０　Ｃ	

審査請求　未請求　請求項の数６　ＯＬ　（全 12 頁）

(21)出願番号　特願平5－327352

(22)出願日　平成５年(1993)12月24日

(71)出願人　000005108
株式会社日立製作所
東京都千代田区神田駿河台四丁目６番地

(72)発明者　前田　章
神奈川県川崎市麻生区王禅寺1099番地　株式会社日立製作所システム開発研究所内

(72)発明者　芦田　仁史
神奈川県川崎市麻生区王禅寺1099番地　株式会社日立製作所システム開発研究所内

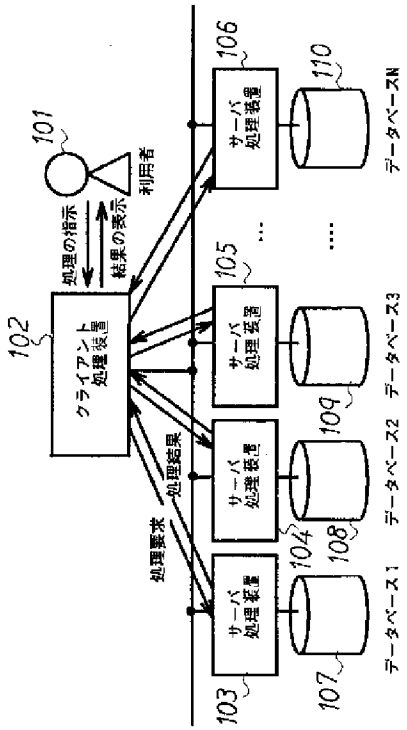
(74)代理人　弁理士　磯村　雅俊

(54)【発明の名称】　データ加工システム

(57)【要約】

【目的】　複数のデータベースに分割して格納されたデータに対して加工処理を行う際に、複数のデータベースに設けられた処理装置を並列に活用することにより高速な処理を実現し、かつ複数のデータベースを接続するネットワークの負荷を軽減し、かつ利用者にとってデータの物理的な配置を意識させずにデータ加工処理を実行することを可能にすること。

【構成】　クライアント処理装置１０２から複数のサーバ処理装置１０３～１０６に分類処理要求を送出し、それを受けた複数のサーバ処理装置は、分類処理を並列に実行し、処理結果をクライアント処理装置に伝達する。処理結果を受けたクライアント処理装置は、それらを合成し、合成した結果を用いてルールインダクションなどの処理を実行する。



【特許請求の範囲】
【請求項１】　クライアント処理装置と、該クライアント処理装置に接続され、それぞれがデータベースに接続された複数のサーバ処理装置からなるデータ加工処理システムにおいて、
処理対象となる１つ以上のテーブル形式のデータを上記データベースの各々に格納する格納手段と、
上記テーブル形式のデータに含まれるデータ項目の値を分類するための分類規則を指定する分類規則指定手段と、
上記分類規則指定手段により指定された分類規則を上記サーバ処理装置に伝達する第１の伝達手段と、
上記サーバ処理装置において、上記伝達された分類規則にしたがってそれぞれのデータベースに格納されているデータに対して分類処理を実行する分類実行手段と、
上記分類実行手段によって分類された結果を上記クライアント処理装置に伝達する第２の伝達手段と、
上記クライアント処理装置に伝達された分類結果を合成して上記処理対象であるテーブル形式のデータに対する分類結果を生成する手段と、
上記生成された分類結果を用いて、データ項目間の関係を分析する分析手段と、
上記分析手段によって分析された結果を出力する出力手段とからなることを特徴とするデータ加工システム。
【請求項２】　請求項１記載のデータ加工システムにおいて、上記複数のサーバ処理装置は同時並列に処理を行うことを特徴とするデータ加工システム。
【請求項３】　請求項１または２記載のデータ加工システムにおいて、上記分類規則は、実数または整数の値を持つデータ項目に対しては上限値および下限値を指定し、文字列としての値を持つデータ項目に対しては文字列の変換を指定するものであることを特徴とするデータ加工システム。
【請求項４】　請求項１～３記載のデータ加工システムにおいて、上記分類規則には、テーブル形式のデータの各レコードを一意に識別する項目が含まれることを特徴とするデータ加工システム。
【請求項５】　請求項１～４記載のデータ加工システムにおいて、上記分析されたデータ項目間の関係を、１つまたは複数のルールとして出力することを特徴とするデータ加工システム。
【請求項６】　請求項１～４記載のデータ加工システムにおいて、上記分析されたデータ項目間の関係を、分類木として出力することを特徴とするデータ加工システム。
【発明の詳細な説明】
【０００１】
【産業上の利用分野】本発明は、物理的に複数のデータベースに分割して格納された数値または記号で表現された情報の集まりを加工し、利用者にとって有用な表現に変換して出力するデータ加工システムに関し、特に、高速な処理を実現し、かつ利用者にデータが物理的に分割されて格納されていることを意識させずに処理を実行することができるデータ加工システムに関する。
【０００２】
【従来の技術】最近の情報処理技術にはめざましいものがあり、電子計算機内に蓄積されるデータ量や処理すべきデータ量は年々増大しており、特にネットワーク化が進むにつれて、オンラインシステムを中心にこの傾向はますます顕著になってきている。現在では、そのデータ量はギガバイト（＝１０の９乗）、レコード数にして１００万件を超えるものも珍しくない。一般に、電子計算機内に蓄積されたデータそのものは単なる数値や記号の集合に過ぎず、そのままでは利用することができない。そこで、データの有効活用を図るために、このデータの集合を有用な情報に変換して利用者に提供するための技術がいろいろ提案されている。その中の代表的な技術として、
（１）回帰分析や重相関分析といった統計的な手法（以下、従来技術１という）
（２）ニューロモデルやファジィモデルを用いて、データ間の相互関係を学習させる方法（以下、従来技術２という）
（３）ルールインダクションなど、知識獲得手法を用いる方法（以下、従来技術３という）
などが従来から公知である。
【０００３】上記従来技術１に関しては、広く一般的に知られている周知の手法であり、ここで詳しく述べることはしない。上記従来技術２に関して、ニューロモデルの学習に関しては多くの文献に記載されているが、これは基本的には多入力多出力の非線形関係を学習によりモデル化するものである。また、非線形関係の学習という意味では、入出力関係をファジィモデルで表現し、ニューロと同様なアルゴリズムを用いて学習によりメンバシップ関数の形状を調節する方法が知られている。ニューロまたはファジィいずれの方法にしても、入出力関係をあるパラメータを含むモデルで表現しておき、そのパラメータを学習により決めることによりモデルを構築するものである。
【０００４】上記従来技術３に関しては、例えば、Springer-Verlag の「Machine Learning」,463～482ページに記載された“Learning Efficient Classification Procedures and Their Application to Chess End Games”というJ.Ross Quinlan の論文がある。この論文には、データから分類のための決定木を自動的に作成するＩＤ３というアルゴリズムが述べられている。また、ルールインダクション手法として、日立クリエイティブワークステーション２０５０マニュアル、「ＥＳ／ＴＯＯＬ／Ｗ－ＲＩ解説／操作」の第２３頁～第５３頁にデータ間に存在する関係をルールの形で表現

する手法が詳細に説明されている。

【０００５】次に、ＩＤ３による決定木作成例を説明しておく。まず、全体の事例の集合Ｃは、

Ｃ＝｛short, blond, blue, ：＋，short, dark, blue：－，tall, dark, brown：－，tall, blond, brown：－，tall, dark, blue：－，short, blond, brown：－，tall, red, blue：＋，tall, blond, blue：＋｝

で与えられているものとする。各事例は身長（値はshort, tall）、髪の色（値はblond, dark, red）、目の色（blue, brown）という３つの属性を持ち、さらに＋とーという２つのクラスの何れかに属している。ＩＤ３は、どの属性をどういう順番で判定すれば、事例を＋またはーというクラスに分類できるかという問題を、決定木を生成することにより解決しようとする。決定木の生成は、どの属性から判定すれば最も判定後の情報量が最小になるか（すなわち、どの属性による判定が最も多くの情報量を持つか）という基準で属性を選んでいく。図８に、上の事例の場合に得られる決定木の例を示す。この決定木の意味するところは次のようなものである。まず、

（ａ）事例を髪の色で分類せよ。

（ａ１）髪の色がdarkならばーである。

（ａ２）髪の色がredならば＋である。

（ａ３）髪の色がblondならば、次に目の色で分類せよ。

（ａ３１）目の色がblueならば＋である。

（ａ３２）目の色がbrownならばーである。

これまで、いろいろなＩＤ３の変形アルゴリズムが提案されているが、基本的にはある基準にしたがって判定の順序を決定するものである。

【０００６】上記従来技術３は、上記従来技術１および２と異なり、データを加工した結果がルールインダクションの場合にはルールとして、またＩＤ３の場合には決定木として、陽に表現された形で得られることから、蓄積されたデータに含まれている関係を利用者が発見するのに特に有効である。これらの技術は大量データの有効利用技術として用いられることが多くなっている。

【０００７】上記のようなデータの利用技術と並行して、大量のデータを高速に検索することを主目的とした並列データベース技術の開発が進んでいる。このような並列データベース技術は、例えば、「日経エレクトロニクス１９９３年７月１９日号」の第９１頁～第１０６頁に記載された“並列マシン向けＤＢＭＳ技術－９０年代半ばの実用化めざす”という論文に解説されている（以下、従来技術４という）。また、利用者のインタフェースとして表計算ソフトを用いてデータベースの検索を行うソフトウェア技術が、例えば、「日経コンピュータ１９９３年７月１２日号」の第６５頁～第７５頁に記載された“サーバのＲＤＢを直結－基幹データを表計算ソフトへ”と題する論文に解説されている（以下、従来技術５という）。この論文に記載されているものでは、サーバ処理装置と、利用者が直接使用するクライアント処理装置をネットワークで接続し、クライアント処理装置では表計算ソフトが提供するインタフェースを用い、データベースの検索要求が発生した場合にはＳＱＬ（Ｓtructured Ｑuery Ｌanguage）などで記述された検索命令をサーバ処理装置に送り、サーバ処理装置がその要求にしたがって検索を実行し、検索結果をクライアント処理装置に転送している。クライアント処理装置はサーバ処理装置から転送された検索結果を表計算の形に整形して利用者に対して表示する。このような仕組みにより、少なくとも検索や簡単な統計量の算出（平均値や分散など）に関しては、利用者は全く分散データベースを意識せずに、全てデータが手元（クライアント処理装置）にあるかのように作業することができる。

【０００８】

【発明が解決しようとする課題】上記従来技術１および２は、電子計算機内に蓄積されたデータを回帰分析、統計処理または学習処理によって加工する技術であるが、加工した結果を利用者がどのように利用するかなど、その利用方法については特に考慮されていなかった。それに対して、従来技術３のルールインダクションやＩＤ３アルゴリズムなどの知識獲得手法を用いる方法では、加工された結果は利用者にとって有用なものになりうるが、この技術は主にどのようなデータ加工手段を用いれば有用な情報を得ることができるかという点に関するものであり、従来技術４で扱っているような並列データベースの上でどのように実行するかという点については、これまで検討されることがなかった。

【０００９】一方、従来技術４の並列データベース技術は、いわゆるオンラインシステムにおけるトランザクション処理と、オフライン処理におけるデータ検索処理を、いかに並列ハードウェアを用いて高速に実行するかに主眼がおかれたものであり、上記従来技術１～３のように大量のデータを高度に加工するような処理を、ハードウェアの並列性を生かして高速に実行するような手段についてはこれまで知られていない。したがって、従来技術３の知識獲得手法を従来技術４の並列データベースに適用することによる生じる効果は、単に、データの検索の高速化という並列データベース本来の効果に限られていた。このように、従来は、ハードウェアが本来持っている並列能力を生かした方法については知られておらず、したがって、従来技術３の知識獲得手法を十分高速に実行することができないという問題点があった。本発明の第１の目的は、上記の問題点を解決し、並列・分散データベース上で高速に実行可能なデータ加工方法および装置を提供することにある。

【００１０】また、一般に、新規なデータを抽出収集して蓄積するのには多大な手間、時間、およびコストがかかるため、既存のデータ資産をいかに有効に利用するかが重要な技術的課題となっている。例えば、同一の会社

の中でも、異なる部門毎にそれぞれ別個の情報をまったく別のデータベースとして構築していることがある。この場合、データベースは物理的にも遠く離れ、まったく別種のハードウェア上に、まったく別種のソフトウェアを用いて構築されていることさえある。上記の従来技術３においては、データは処理を実行する処理装置上に存在すると仮定していた。すなわち、従来技術３を適用するためには、分散したハードウェア上にある関連データを切り出して、１つの処理装置上に集めてくる処理が必要になる。現在では、処理装置はネットワークで接続されていることが多いが、それでも関係するデータを集めたデータベースをデータ加工を試みる毎に定義して構成するというのは、そのために時間を要し、また、ネットワークの負荷としても、また人的作業の点からも、非常にコストがかかるという問題点があった。上記従来技術３を利用する側から見た場合、利用者は様々な目的を達成するためにデータ加工を行うのであって、そのためのデータの準備に多くの時間、コストをかけるというのは非効率的であり、さらに、これらの技術の有効性を十分に生かしきれていないという問題点があった。

【００１１】上記問題点を、データベース検索の場合に解決しようとする技術が上記従来技術５である。しかしながら、この従来技術５の対象とするのは、あくまでも通常の表計算で用いるような処理だけであり、上記従来技術３で対象としているような高度なデータ加工処理に対しては対策がなされていなかった。例えば、ルールインダクションによるルール抽出処理を実行しようとすると、クライアント処理装置上の処理でデータを参照するたびにサーバ処理装置へのデータ要求が発生するため、ネットワークの負荷は非常に重くなり、ほとんど実用的な処理速度にはならないという問題点がある。特に、データベースが遠く離れた場所にある場合、例えば、あるデータベースが東京に、また別のデータベースが大阪に存在するような場合には、この問題点はより顕著になって現れる。本発明の第２の目的は、上記の問題点を解決し、利用者にとっては物理的なデータの存在場所を意識することなく、ネットワークで接続されたサーバ処理装置上のデータを用いて、ルールインダクションなどの手法を用いてデータ間の関係を利用者にとって有用な形に加工することができ、またこれらの処理をネットワークに接続された処理装置の処理能力を有効に活用することにより、高速に実行する方式および装置を提供することにある。

【００１２】

【課題を解決するための手段】本発明は、上記の問題点を解決するために、クライアント処理装置と、該クライアント処理装置に接続され、それぞれがデータベースに接続された複数のサーバ処理装置からなるデータ加工処理システムにおいて、処理対象となる１つ以上のテーブル形式のデータを上記データベースの各々に格納する格納手段と、上記テーブル形式のデータに含まれるデータ項目の値を分類するための分類規則を指定する分類規則指定手段と、上記分類規則指定手段により指定された分類規則を上記サーバ処理装置に伝達する第１の伝達手段と、上記サーバ処理装置において上記伝達された分類規則にしたがってそれぞれのデータベースに格納されているデータに対して分類処理を実行する分類実行手段と、上記分類実行手段によって分類された結果を上記クライアント処理装置に伝達する第２の伝達手段と、上記クライアント処理装置に伝達された分類結果を合成して上記処理対象であるテーブル形式のデータに対する分類結果を生成する手段と、上記生成された分類結果を用いて、データ項目間の関係を分析する分析手段と、上記分析手段によって分析された結果を出力する出力手段とを有している。

【００１３】

【作用】本発明は、上記各手段、特に、テーブル形式のデータに含まれるデータ項目の値を分類するための分類規則を指定する分類規則指定手段と、分類規則指定手段により指定された分類規則を上記サーバ処理装置に伝達する第１の伝達手段と、サーバ処理装置において上記伝達された分類規則にしたがってそれぞれのデータベースに格納されているデータに対して分類処理を実行する分類実行手段と、分類実行手段によって分類された結果をクライアント処理装置に伝達する第２の伝達手段と、クライアント処理装置に伝達された分類結果を合成して処理対象であるテーブル形式のデータに対する分類結果を生成する手段と、生成された分類結果を用いてデータ項目間の関係を分析する分析手段とを有することによって、時間のかかる処理を並列に実行することができ、かつ利用者には並列処理を全く意識させることなく、また大量のデータ自身を転送することがなく、目的の処理を実行することができる。また、ルールインダクションに適用した場合、ルールインダクションの処理自身はクライアント処理装置上で実行されるが、クライアント処理装置上には各データベース上で分類された結果だけを保持していればいいので、ネットワークの負荷を増大させることなく、目的の処理を実行することができる。さらに、利用者は使用しようとするデータが物理的にどのデータベース上に存在するかということは全く意識することなく処理を実行し、かつ適切な処理結果を得ることができる。

【００１４】

【実施例】まず、本発明の実施例の概略を説明する。図１は本発明を実施するための全体構成である。図１において、１０１は本発明のデータベース加工装置を利用する利用者、１０２はクライアント処理装置、１０３～１０６はサーバ処理装置、１０７～１１０はそれぞれサーバ処理装置１０３～１０６に接続されているデータベース、１１１はクライアント処理装置とサーバ処理装置を

接続するバスである。複数のデータベース１０７～１１０は互いに独立なＤＢＭＳ（Date Base Management System）で管理されている場合であっても、全くネットワーク上で統一した管理を行う並列ＤＢＭＳで管理されている場合であってもよい。ここでは前者の場合を例にとって説明する。また、説明の便宜上、データは図２に示したように複数の属性からなる複数のレコード（行に対応）によって記述されており、それら複数のレコードは複数のデータベースに分割されて格納されているものとする（図２では、それぞれK個の属性を有するM個のレコードが３個のデータベースに分割されて格納されることを示している）。

【００１５】例えば、クライアント処理装置１０２では、複数のデータベース１０７～１１０に分散されて格納されているデータに対して、仮想的に一つのまとめ上げられた表のイメージで表示することができる（従来技術５）。このとき、利用者は、その表イメージのデータに対して、例えば、ルールインダクションによる分類ルールの抽出を行うものとする。複数のデータベース上に蓄積されているデータは一般に膨大なものであるから、クライアント処理装置に全てのデータを転送することは処理時間とコストの点から一般には行うことができない。

【００１６】そこで、本発明では、クライアント処理装置上で動作するルールインダクション処理が、ある属性または属性値による分類結果を求める処理要求を複数のサーバ処理装置に送信する。処理要求の形式は、例えば、一般によく知られているＳＱＬ（Structured Query Language）文の形式でもよい。複数のサーバ処理装置は、クライアント処理装置から要求された処理をそれぞれ並列に実行し、処理結果をクライアント処理装置に伝達する。ルールインダクション処理の場合、この処理結果は、クライアント処理装置によって指定された属性または属性値によって各データベース中のレコードを分類した結果を、例えば、レコード番号と分類結果の対応表の形で表現されている。クライアント処理装置は、複数のサーバ処理装置から伝達された処理結果をまとめ上げることにより、以後のルールインダクションに必要な情報、すなわちそれぞれの属性または属性値による全レコードの分類表をクライアント処理装置上に作成することができる。したがって、データベースに蓄積された情報自身をクライアント処理装置に転送することなしに、分類結果だけをクライアント処理装置上にもつことによって、ネットワークの負荷を軽減し、高速に処理をすることが可能になる。

【００１７】また、このような処理を行わずに、クライアント処理装置が直接ルールインダクションを実行した場合には、ルールインダクション処理の進行に従って複数のサーバ処理装置への検索要求が逐次伝達されることになり、やはりネットワークの負荷が非常に大きくなり、結果としてルールインダクション処理に要する時間が多大なものになってしまう。本発明で設けた手段により、このようなオーバヘッドを低減できるので、高速な処理を実現することができる。

【００１８】以下、本発明の実施例の詳細な動作を説明する。実施例の説明にあたって、まず、ルールインダクションの動作原理について説明しておく。詳細については、例えば、上述した文献、日立クリエイティブワークステーション２０５０マニュアル、「ＥＳ／ＴＯＯＬ／Ｗ－ＲＩ解説／操作」第２３頁～第５３頁に解説されている。図３はルールインダクションの入力となる事例テーブルの例である。ルールインダクションの入力となる事例テーブル２００において、各行がそれぞれ一つの事例に対応している。この例では、クレジットの申し込みがそれぞれ一件の事例に対応する。それぞれの事例は１つまたは複数の属性をもっている。図３の例では、「銀行の信用」２０１、「預金残高」２０２、「現在の負債」２０３、「クレジット査定」２０４という４つの属性がある。それぞれの属性は、あらかじめ定められた形式の属性値をもつ。図３の例では、「銀行の信用」２０１という属性は「有」と「無」という２つの属性値のいずれかの値をもつ。このような属性を「記号属性」と呼ぶ。それに対して「預金残高」２０２と「現在の負債」２０３という属性は数値で表される属性、すなわち、数値属性をもっている。一般の事例テーブルでは、記号属性と数値属性が混在する。また、図３の例において、事例の中にはその属性の一部が不明なものも存在する（参照符号２０５参照）。これを「欠損値」と呼ぶ。

【００１９】ルールインダクションの実行にあたっては、まずどの属性を結論属性とするかを指定する。図３の例では、「クレジット査定」を自動的に判別するシステムを作ることを目的として、「クレジット査定」という属性が結論属性として指定されるものとする。一般のルールインダクション処理では結論属性は記号属性でなければならない。さらに、各事例における結論属性の属性値を説明するために入力として扱われる説明属性を指定する。図３の例では、結論属性以外の属性をすべて説明属性として扱うものとする。もちろん、事例テーブル中の一部分だけを説明属性として指定することもできる。

【００２０】図３のような事例テーブルにおいて、事例の数（行の数）が大きくなるにつれて、数値属性をそのまま扱うことが困難になってくる。そこで、数値属性を記号属性に置き換える「ラベル付与処理」をルールインダクション処理に先立って行うことがある。ラベル付与処理とは、数値属性をもつ属性に対して、例えば、適当な区間分割を行って、各区間にラベルと呼ばれる記号を割り当てることによって数値を記号に置き換える処理のことを指す。図４はこのようなラベル付与処理を行ってすべての属性を記号属性に置き換えた事例テーブルの例

である。これは、「預金残高」２０８が１０００万円以上を「大」、１０００万未満で正のものを「中」、預金残高マイナスのものを「小」とした例である。「現在の負債」に関しては、８０００万円以上を「大」、８０００万円未満かつ４０００万円以上のものを「中」、４０００万円未満のものを「小」としてラベル付与した場合である。

【００２１】このように、実数または整数の値を持つデータ項目に対してはそれぞれ上限値および下限値を指定することによっていくつかのグループに分類することができる。また、上限値および下限値を指定する際に、各グループが同じ数、ほぼ同じ数になるように分割するようにしてもよい。ラベル付与処理によって、数値属性の細かな差を無視し、大きく分類した記号属性値にしたがって属性間の関係を解析することができる。特に事例数が大きな事例テーブルに対しては、処理時間の点でも出力される分類ルールの有用性という点でも有効である。また、文字列としての値を持つデータ項目に対しては、予めいくつかの文字列を決めておき、そのいずれかの文字列に変換することによって適当な数のグループに分類することができる。

【００２２】次に、ルールインダクション処理では、結論属性として指定された属性の属性値を、できるだけうまく説明するような説明属性値の組を見つけ、それをルールとして抽出する処理を行う。この処理にはさまざまなオプションがあり、それらのオプションを適当に設定することにより、利用者にとってより有用なルールを抽出することができる。ここではこれらのオプションについては説明しない。詳細は上記文献、日立クリエイティブワークステーション２０５０マニュアル、「ＥＳ／ＴＯＯＬ／Ｗ－ＲＩ解説／操作」の第２３頁～第５３頁に記載されている。

【００２３】図５（ａ）、（ｂ）は、ルールインダクション処理によって抽出されたルールの一例である。ルールインダクションの内部処理についてはさまざまな方式が提案されているが、基本的には、１つまたは複数の属性が特定の属性値（または属性値の組）をもつ事例の数をベースにして、結論属性を最もよく分類する属性値の組を見つける処理からなる。したがって、ルールインダクション処理を実行するに当たっては、各事例の各属性がどの記号属性値をもつかという対応表があれば十分であることが分かる。すなわち、ルールインダクションの初期処理において、このような対応表を準備しておけば、それ以後の処理は全てこの対応表だけに基づいて実行することができる。

【００２４】図６にこのような対応表２１１の例を示す。図６の対応表の各行は、図３の事例テーブルと同様にそれぞれが一つの事例に対応する。図６の各列は、それぞれ属性「銀行の信用」２１３、「預金残高」２１４、「現在の負債」２１５、「クレジット査定」２１６に対応している。対応表２１１の各成分は、対応する事例の対応する属性値がどの値をもつかを表すラベルコードを示している。図６では簡単のため、事例を区別するためにユニークな事例番号（レコード番号）２１２を付与してある。また、対応表２１１の各成分の値がどの記号属性値に対応するかは、図７に示したような別の対応表で管理することができる。図７において、（ａ）は銀行の信用の各成分（無／有）と記号属性値（０／１）の対応表、（ｂ）は預金残高の各成分（小／中／大／不明）と記号属性値（０／１／２／３）の対応表、（ｃ）は現在の負債の各成分（小／中／大）と記号属性値（０／１／２）の対応表、（ｄ）はクレジット査定の各成分（不可／可）と記号属性値（０／１）の対応表である。

【００２５】ここで重要なのは、各属性に対する属性値が何種類の値をとるかに応じて、ラベルコードのとる値は一般に数ビットで表現できることである。図３の事例テーブル自身は、例えば、「クレジット査定」の属性に対して、“不可”という文字を表すために４バイト（全角漢字２つ分）のデータ量を割り当てる必要がある。それに対して、図６の対応表のラベルコードは“１”と“０”を識別できるだけでよいから、基本的には１ビットでよい。したがって、ルールインダクションを非常に大量の事例に対して実行する場合、図３または図４の事例テーブルから図６の対応表にあらかじめ変換しておくことにより、効率のよい処理が可能になる。以上、ルールインダクション処理の概要について説明した。ＩＤ３などの分類の決定木を作成するアルゴリズムも、上記の対応表の参照だけで実行できることはいうまでもない。

【００２６】さて以上の処理は、本発明の手段を用いることによって、図１に示したような環境で実行することができる。すなわち、図１の構成では、ルールインダクション処理を実行するクライアント処理装置と、データの存在するサーバ処理装置が物理的に異なる処理装置であることと、またデータ自身も複数のサーバ処理装置に分散して格納されているから、上記の処理をそのまま単純に実行することはできない。全ての関係するデータをクライアント処理装置上に転送し、クライアント処理装置上で処理を集中的に行うことができれば話は簡単であるが、特に大量のデータを扱う場合や、データベース自身が物理的に遠く離れた場所にあってデータ転送に大きなコストがかかる場合には実際的ではない。

【００２７】以下、本発明の第１実施例を詳細に説明する。第１実施例は、図１および図２の構成においてルールインダクション処理を高速かつ効率的に実行するものである。図８に事例テーブル２２０の構成例を示す。この例では、事例を区別する「事例番号」２２１、各事例の属性値として、「氏名」２２２、「氏名コード」２２３、「銀行の信用」２２４、「預金残高」２２５、「年収」２２６、「現在の負債」２２７、「年齢」２２８、「性別」２２９、「住所」２３０、「配偶者の有無」２

３１、「扶養家族人数」２３２、「職業」２３３、「最終学歴」２３４、「住居の種類」２３５、「クレジットカード保有枚数」２３６、「利用実績」２３７、「クレジット査定」２３８からなっている。

【００２８】図９に、図８に示した事例テーブル２２０を複数のデータベースへの分割して格納する方法を示す。本実施例では、図９のように事例番号１～１０００はデータベース１に、事例番号１００１～２５００はデータベース２に、事例番号２５０１～４５００はデータベース３に、というように事例テーブルを行（レコード）で分割して格納するものとする。各サーバ処理装置は、ローカルにどの範囲のレコードをもっているかを管理しており、クライアント処理装置からの検索処理要求を受けると、ローカルなデータの範囲でだけ検索を実行して結果をクライアント処理装置に伝達する機構を設けている。この方法は、並列ＤＢＭＳでは「shared nothing」方式と呼ばれているものである。もちろん、並列ＤＢＭＳのように全体を統一的に管理するＤＢＭＳが存在せず、単に独立したローカルなＤＢＭＳでそれぞれのデータベースが管理されている場合でも、以下の処理は同様である。

【００２９】全体処理の流れを図１０に示す。まず、ステップ２５０において、クライアント処理装置（図１参照）は、図８に示された形式で事例テーブルの一部を表示装置に表示する。利用者は表示装置上で、結論属性と説明属性を指定する（ステップ２５１）。ここで、一般的に考えて明らかに結論属性と関係がない属性は省いておく。この例では、「氏名」２２２、「氏名コード」２２３、「住所」２３０などの属性は結論属性「クレジット査定」と関係がないと考えられるから予め省いておく。ここではそれ以外の全ての属性を説明属性として指定した場合を考える。

【００３０】次のステップ２５２において、クライアント処理装置は、説明属性および結論属性に対して、ラベル付与処理に必要な情報である「ラベルコード対応表」を作成する。図１１にこのラベルコード対応表２６１の例を示す。ラベルコード対応表２６１において、参照符号２６２は、「残高」という属性に対して定義された「小」というラベルは属性値が０未満の値に対応し、「中」というラベル値は属性値が０以上１０００未満に対応し、「大」というラベル値は属性値が１０００以上に対応していることを意味している。ラベルコード対応表２６１中で値が指定されていない部分は制約が指定されていないものと考える。また、欠損値に対応する「不明」というラベル値に対しても、特別のラベルコード値を割り当てるものとする。例えば、ラベルコード対応表２６１中に参照符号２６３で示したように、残高不明に対してラベルコード“３”を対応させている。また、参照符号２６４、２６５に示したように、記号属性に関しても、属性値とラベルコードの対応関係を指定する。

【００３１】次に利用者によってルールインダクション処理の開始が指示される（ステップ２５３）と、ステップ２５４において、クライアント処理装置は、複数のサーバ処理装置に対し、図１１で示した情報からなる検索要求を送信する。

【００３２】ステップ２５５において、各サーバ処理装置（図１参照）は図１１の検索要求を受信すると、それぞれのサーバ処理装置において検索処理を開始する。まずそれぞれの事例データを、図１１のラベルコード対応表に基づいて、各事例を図６と同様の事例－属性値対応表に変換する（ステップ２５６）。図１２にその一例を示す。この検索処理では、図１１のラベルコード対応表に現れる全ての属性に対して、その属性が記号属性であれば属性値を指定されたラベルコードに置き換え、数値属性であればラベルコード対応表に指定された数値範囲で分類してラベルコードに置き換える処理を行う。各サーバ処理装置は、ローカルに管理している事例テーブルを図１２の事例－属性値対応表に変換した後、ステップ２５７において、その結果を検索結果としてクライアント処理装置に送信する。

【００３３】クライアント処理装置は、複数のサーバ処理装置から処理結果を受信し（ステップ２５８）、それを合成して、事例テーブル全体の事例－属性値対応表をクライアント処理装置上に作成する（ステップ２５９）。この処理は、図９に示した事例テーブルの分割と全く逆に、それぞれのサーバ処理装置からの検索結果を行方向に結合すればよい。この事例－属性値対応表が作成された後は、ルールインダクション処理はこの表だけを参照してルールを抽出する（ステップ２６０）。以上の説明は、ルールインダクションに限らず、前述したＩＤ３のような属性に基づいて分類の決定木を自動生成する処理にも全く同様に適用することができる。

【００３４】以上、本発明の第１の実施例によれば、クライアント処理装置におけるルールインダクション処理の実行に当たって、あらかじめ複数のサーバ処理装置で実行されるＤＢＭＳの機能を拡張しておき、クライアント処理装置で必要な検索結果をクライアント処理装置からの要求に従って並列に実行させることが可能になる。複数のサーバ処理装置への検索要求とその結果の取得は、ルールインダクション処理本体の実行に先立って一度だけ実行すればよく、ネットワーク上で流れるデータ量を大幅に削減し、ネットワーク負荷を減少させることが可能になる。また、検索処理自身も並列に実行されるため、全体の処理時間を短縮できるという効果もある。さらに利用者から見れば、あくまでクライアント処理装置上の表イメージのデータに対して直接処理を実行し、結果を得るという作業が可能になるため、事例テーブルが複数のデータベースに分割して格納されていることは全く意識することがなく、より自然に作業を進めることができるという効果もある。

【００３５】本発明の第１の実施例では、図２に示すように、事例テーブルを行単位に分割して複数のデータベースに格納することを前提としていた。すなわち、複数のデータベースは、あらかじめ組となって一つの事例テーブルを格納するために使用されていたということができる。ところが実際には、複数の部署が全く独立にデータベースを構築し、それらのデータベースを統合して利用する場合がある。その場合には、事例テーブルの分割は、行単位ではなく、列単位に分割されていると考えることができる。

【００３６】本発明の第２の実施例は、列単位で分割された複数のデータベース上のデータを用いてルールインダクションなどの処理を行うものである。図１３に、本実施例におけるデータベースの構成を示す。データベースの内容は第１の実施例で説明した図８の構成と同様であるが、家族情報と金融関連情報が別個のデータベースに格納されている。図１３（ａ）は家族情報に関するデータベース３００であり、「氏名」３０１、「氏名コード」３０２、「年齢」３０３、「性別」３０４、「住所」３０５、「配偶者の有無」３０６、「扶養家族人数」３０７、「職業」３０８、「年収」３０９、「最終学歴」３１０、「住居の種類」３１１という属性からなっている。また、図１３（ｂ）は金融関連情報データベース３２０であり、「氏名」３２１、「氏名コード」３２２、「銀行の信用」３２３、「預金残高」３２４、「年収」３２５、「職業」３２６、「クレジットカード保有枚数」３２７、「利用実績」３２８、「クレジット査定」３２９という属性からなっている。

【００３７】これらのデータベースは、当初は全く異なる目的のために構成されたものであってもよく、またハードウェア的にも全く異なる装置上に存在していてもよい。さらには、物理的に遠くはなれた場所に存在していてもかまわない。ただし、これらのデータベースはネットワークに接続されていて、利用者が直接使用するクライアント処理装置からオンラインで利用できなければならない。ここで、２つのデータベースに「氏名」、「氏名コード」、「職業」、「年収」という属性が共通して存在する。このうち「氏名コード」をレコードを特定するためのキー属性と考える。２つのデータベースの整合性を管理していない場合は、「職業」や「年収」という属性の値は、２つのデータベースで異なることもありうる。そこで、利用者は図８のテーブルの形でデータを扱う際に、各項目が物理的にどのデータベースのどういう項目に関係づけられているかを指定しなければならない。

【００３８】図１４にこの関係を定義する属性関係づけテーブル３３０の例を示す。図１４のテーブル３３０は、属性、データベース種別、項目名から構成され、説明属性・結論属性・キー属性などの各属性がそれぞれどのデータベース中の（データベース種別）どの属性（項目名）に対応しているかを表している。例えば、図１４のテーブル３３０の例では、「銀行の信用」、「残高」という属性は、金融関係データベースの項目名「信用」、「預金残高」に対応させ、「氏名」、「年収」という属性は家族情報データベースの項目名「氏名」、「年収」に対応したものであることを示している。キー属性はレコードを特定するものであるから、当然使用する全てのデータベースに属性として含まれている必要がある。このテーブルはあらかじめ利用者がデータ加工の目的を考えあわせて、どのデータベースのどの属性を用いるかを設計して定義するものとする。

【００３９】この後のルールインダクション処理のフローチャートを図１５に示す。まず、ステップ３４０において、クライアント処理装置（図１参照）は、図８に示された形式で事例テーブルの一部を表示装置に表示する。利用者は表示装置上で、結論属性と説明属性を指定する（ステップ３４１）。ここで、一般的に考えて明らかに結論属性と関係がない属性は省いておく。この例では、「氏名」、「氏名コード」、「住所」などの属性は結論属性「クレジット査定」と関係がないと考えられるから予め省いておく。ここではそれ以外の全ての属性を説明属性として指定した場合を考える。

【００４０】次のステップ３４２において、クライアント処理装置は、説明属性および結論属性に対して、ラベル付与処理に必要な情報である「ラベルコード対応表」を作成する。図１６にこのラベルコード対応表の例を示す。図１６（ａ）は家族情報データベースに対するラベルコード対応表３６０、図１６（ｂ）は金融関連情報データベースに対するラベルコード対応表３７０である。対応表の意味するところは図１１で説明したものと大体同じであるが、キー属性を指定する項目が含まれていることが異なっている。それ以外では、ここまでの処理は第１実施例で説明したものと同様である。ここで、ルールインダクションに用いる属性がそれぞれどのデータベースの属性であるかは、図１４の属性関係づけテーブルを参照して決定することができる。

【００４１】次に利用者によってルールインダクション処理の開始が指示される（ステップ３４３）と、クライアント処理装置は、２つのサーバ処理装置に対し、それぞれ図１６（ａ）、（ｂ）に示したラベルコード対応表３６０、３７０からなる検索要求を送信する。

【００４２】ステップ３４５において、各サーバ処理装置（図１参照）は図１６の検索要求を受信すると、それぞれのサーバ処理装置において検索処理を開始する。まずそれぞれの事例データを、図１６（ａ）、（ｂ）のラベルコード対応表に基づいて、各事例を図６と同様の事例－属性値対応表に変換する（ステップ３４６）。図１７にその一例を示す。この検索処理では、図１６（ａ）および（ｂ）のラベルコード対応表３６０および３７０に現れる全ての属性に対して、その属性が記号属性であ

れば属性値を指定されたラベルコードに置き換え、数値属性であればラベルコード対応表に指定された数値範囲で分類してラベルコードに置き換える処理を行う。ここで、第１の実施例における図１２との違いは、複数のデータベースには事例テーブルが列方向に分割されて格納されているため、図１７の事例－属性値対応表にはレコードに固有な識別情報として指定されたキー属性である「氏名コード」の情報が付加されていることである。各サーバ処理装置は、ローカルに管理している事例テーブルを図１７の事例－属性値対応表に変換した後、ステップ３４７において、その結果を検索結果としてクライアント処理装置に送信する。

【００４３】クライアント処理装置は、複数のサーバ処理装置から処理結果を受信し（ステップ３４８）、レコード識別情報である「氏名コード」情報をキーとして合成して、事例テーブル全体の事例－属性値対応表をクライアント処理装置上に作成する（ステップ３４９）。

【００４４】ここで重要なのは、２つのデータベースの整合性は管理されていない場合があるので、どちらか一方の事例テーブルにしか存在しないレコードがありうることである。このような場合、合成された事例－属性値対応表の中で対応しない部分は欠損値を表す特別な値を埋めこむことにすれば、キー属性を基に事例テーブル全体に対する事例－属性値対応表を合成することができる。事例－属性値対応表が作成された後は、ルールインダクション処理はこの表だけを参照してルールを抽出する（ステップ３５０）ことは、第１実施例と全く同様である。

【００４５】以上の説明は、ルールインダクションに限らず、ＩＤ３のような属性に基づいて分類の決定木を自動生成する処理にも全く同様に適用することができる。以上述べた本発明の第２実施例によれば、クライアント処理装置におけるルールインダクション処理の実行に当たって、あらかじめ複数のサーバ処理装置で実行されるＤＢＭＳ（データベース管理システム）の機能を拡張しておき、クライアント処理装置で必要な検索結果をクライアント処理装置からの要求に従って並列に実行させることが可能になる。複数のサーバ処理装置への検索要求とその結果の取得は、ルールインダクション処理本体の実行に先立って一度だけ実行すればよく、ネットワーク上で流れるデータ量を大幅に削減し、ネットワーク負荷を減少させることが可能になる。また、検索処理自身も並列に実行されるため、全体の処理時間を短縮できるという効果もある。さらに利用者から見れば、あくまでクライアント処理装置上の表イメージのデータに対して直接処理を実行し、結果を得るという作業が可能になるため、事例テーブルが複数のデータベースに分割して格納されていることは全く意識することがなく、より自然に作業を進めることができるという効果もある。

【００４６】以上の効果は第１実施例と同様であるが、さらに第２実施例特有の効果としては、整合性の保証されていない複数の独立したデータベース上に分割して格納されている情報を、あたかも１つの仮想的なデータベースとして取り扱うことができるという効果もある。データ加工処理をこのように実現することにより、従来の統計解析処理や、学習によるニューロモデルまたはファジィモデルの構築などと全く同様のインタフェースを用いてルールインダクションなどの処理を行うことができる。このように統一されたインタフェースでさまざまな種類のデータ加工処理を自由に実行し、結果を分析することができるので、利用者の作業効率が向上するという効果もある。

【００４７】

【発明の効果】以上、本発明によれば複数のデータベースに分割して格納されているデータに対してルールインダクションなどのデータ加工処理を適用する際に、複数のデータベースが備えているデータ処理能力を並列に活用することができるので、高速な処理が可能になるという効果がある。また、データ加工に用いるデータそのものを所定の装置（クライアント処理装置）上に集める必要もなく、また複数のデータベースに対する検索処理はそれぞれ１回で十分なため、ネットワークの負荷を大幅に軽くすることができるという効果がある。さらに、利用者にとっては全てテーブル形式で仮想的に定義されたデータを対象に処理を進めることができ、実際のデータが物理的に分割して複数のデータベースに格納されていることを意識する必要がないので、データ加工処理を容易に進めることができるという効果もある。

【図面の簡単な説明】

【図１】本発明の対象とする複数データベース利用データ加工処理システムの全体構成図である。

【図２】複数のデータベースへのデータ分割格納方式を説明するための図である。

【図３】事例テーブルの例を示す図である。

【図４】ラベル付与処理により記号属性に変換された事例テーブルの例を示す図である。

【図５】ルールインダクションにより抽出されたルールの例を示す図である。

【図６】事例－属性値対応表の例を示す図である。

【図７】表成分値－属性値対応表を示す図である。

【図８】クレジット査定における事例テーブルの構成図である。

【図９】事例テーブルの分割格納方式を示す図である。

【図１０】本発明の第１実施例の処理フローチャートである。

【図１１】ラベルコード対応表の例を示す図である。

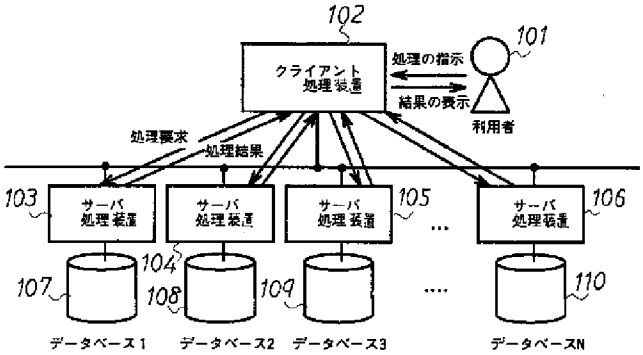
【図１２】サーバ処理装置における事例－属性値対応表の例を示す図である。

【図１３】列分割による事例テーブルの分割格納方式を示す図である。

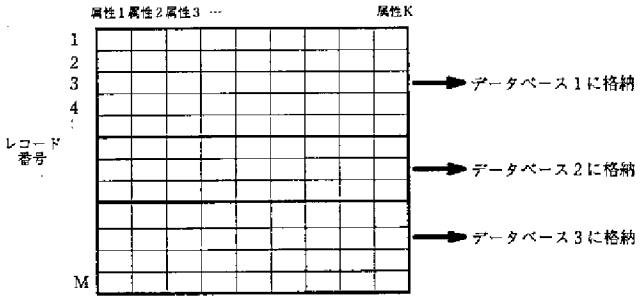
【図１４】属性関連づけテーブルの例を示す図である。
【図１５】本発明の第２実施例の処理フローチャートである。
【図１６】ラベルコード対応表の例を示す図である。
【図１７】サーバ処理装置における事例－属性値対応表の例を示す図である。
【図１８】ＩＤ３によって生成された分類木の例を示す図である。

【符号の説明】
１０１　利用者
１０２　クライアント処理装置
１０３～１０６　サーバ処理装置
１０７～１１０　データベース

【図１】



【図２】



【図３】

201　202　203　204　200

銀行の信用	預金残高	現在の負債	クレジット査定
有	400	2200	可
有	1000	2500	可
無	2000	1000	可
有	-2000	10000	不可
有	400	6000	可
有	100	1000	可
無	400	9000	不可
無	1000	5000	不可
無		9500	不可

【図４】

207　208　209　210　206

銀行の信用	預金残高	現在の負債	クレジット査定
有	中	小	可
有	大	小	可
無	大	小	可
有	小	大	不可
有	中	中	可
有	中	小	可
無	中	大	不可
無	大	中	不可
無	一	大	不可

【図５】

ルール１
ＩＦ　現在の負債＝小
　　預金残高＝中または大
ＴＨＥＮ　クレジット査定＝可

ルール２
ＩＦ　現在の負債＝大
ＴＨＥＮ　クレジット査定＝不可

【図６】

212　213　214　215　216　211

レコード番号	銀行の信用	預金残高	現在の負債	クレジット査定
1	1	1	0	1
2	1	2	0	1
3	0	2	0	1
4	1	0	2	0
5	1	1	1	1
6	1	1	0	1
7	0	1	2	0
8	0	2	1	0
9	0	3	2	0

【図７】

(a)

銀行の信用	
無	0
有	1

(b)

預金残高	
小	0
中	1
大	2
不明	3

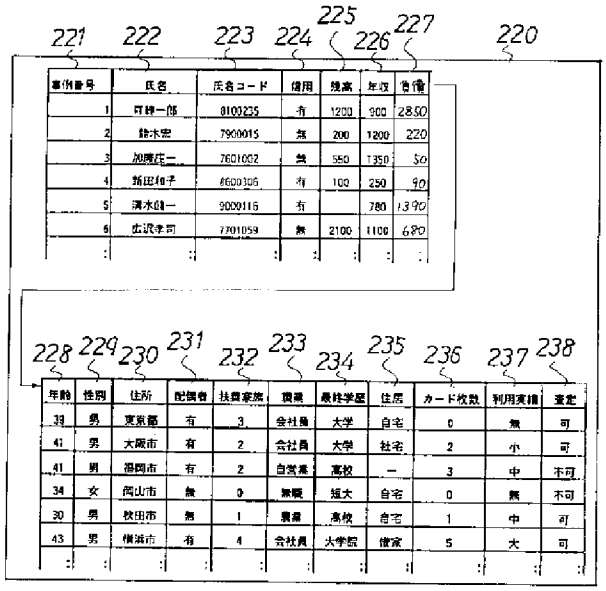
(c)

現在の負債	
小	0
中	1
大	2

(d)

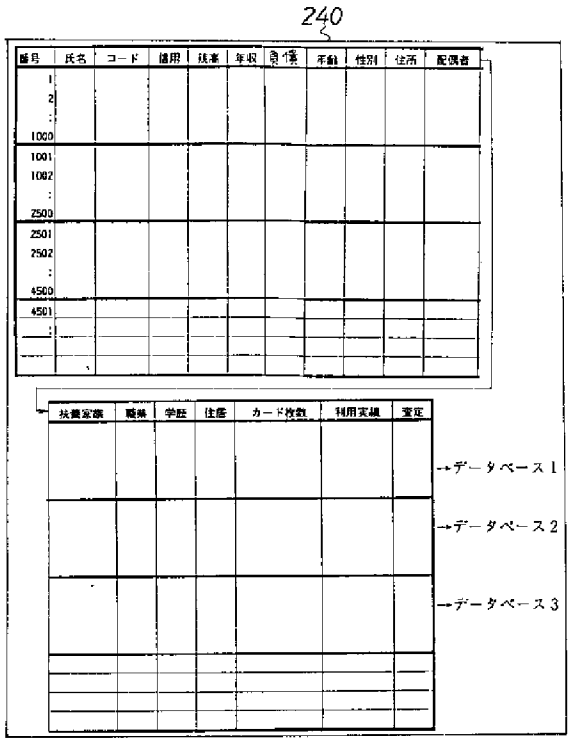
クレジット査定	
不可	0
可	1

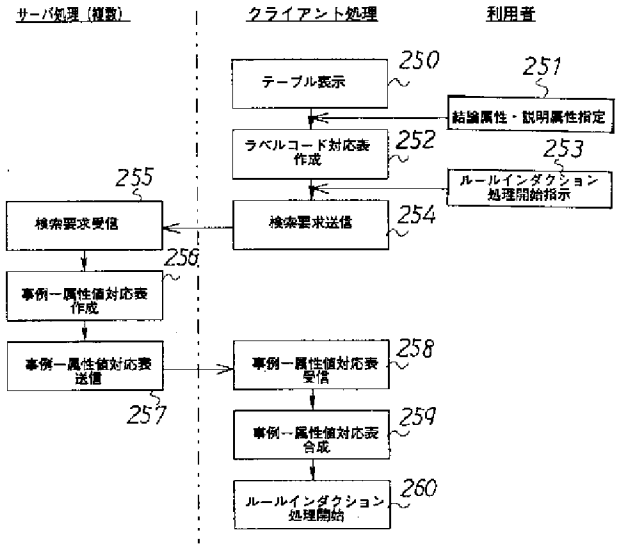
【図８】



【図９】

【図１０】





【図１１】

261

属性値	ラベル値	最小値	最大値
信用：無	0	-	-
信用：有	1	-	-
残高：小	0	-	0
残高：中	1	0	1000
残高：大	2	1000	-
残高：不明	3	-	-
年齢：若い	0	0	20
年齢：青年	1	20	35
年齢：中年	2	35	50
年齢：壮年	3	50	-
性別：男	0	-	-
性別：女	1	-	-
：	：	：	：

264
262
263
265

【図１２】

レコード番号	信用	残高	年齢	性別	配偶者	…
1	1	2	1	0	1	
2	0	1	2	0	1	
3	0	1	2	0	1	
4	1	1	0	1	0	
5	1	3	1	1	1	
6	0	2	1	0	1	
:	:	:	:	:	:	

【図１３】

(a)

321 氏名	322 コード	323 信用	324 残高	325 年収	326 職業	327 カード枚数	328 利用実績	329 審査
:	:	:	:	:	:	:	:	:
森紳一郎	8100235	無	1200	950	会社員	0	無	可
:	:	:	:	:	:	:	:	:

320

【図１４】

330

属性	データベース種別	項目名
氏名	家族情報ＤＢ	氏名
氏名コード	キー属性	氏名コード
銀行の信用	金融関連情報ＤＢ	信用
残高	金融関係情報ＤＢ	預金残高
年収	家族情報ＤＢ	年収
:	:	:

【図１６】

(b)

301 氏名	302 コード	303 年齢	304 性別	305 住所	306 配偶者	307 扶養家族	308 職業	309 年収	310 学歴	311 性歴
森紳一郎	8100235	39	男	東京都	有	3	会社員	900	大学	自宅
:	:	:	:	:	:	:	:	:	:	:

300

【図１５】

(a)

360

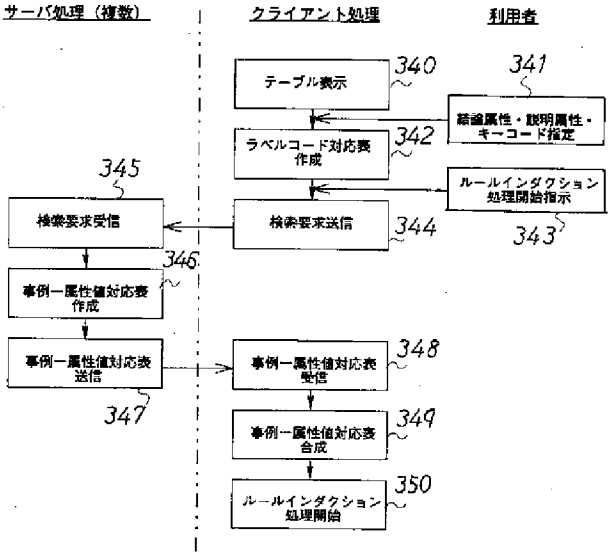
属性値	ラベル値	最小値	最大値
キー属性：氏名コード			
年齢：若い	0	0	20
年齢：青年	1	20	35
年齢：中年	2	35	50
年齢：壮年	3	50	-
性別：男	0	-	-
性別：女	1	-	-
:	:	:	:

(b)

370

属性値	ラベル値	最小値	最大値
キー属性：氏名コード			
残高：小	0	-	0
残高：中	1	0	1000
残高：大	2	1000	-
残高：不明	3	-	-
:	:	:	:

【図１７】



【図１８】

氏名コード	レコード番号	信用	残高	年齢	性別	配偶者	…
8100235	1	1	2	1	0	1	
7900015	2	0	1	2	0	1	
7601002	3	0	1	2	0	1	
8600306	4	1	1	0	1	0	
8000116	5	1	3	1	1	1	
7701059	6	0	2	1	0	1	
:	:	:	:	:	:	: